



H-infinity Based Active Queue Management Design for Congestion Control in Computer Networks[†]

Assist. Prof. Dr. Hazem I. Ali¹, Mr. Karam Samir Khalid²

^{1,2}Control and Systems Engineering Dept., University of Technology

e-mail: Hazem.i.ali@uotechnology.edu.iq, karam_samir1986@yahoo.com

Received: 8/06/2014

Accepted: 11/09/2014

Abstract – In this paper the design of robust Active Queue Management (AQM) for congestion control in computer networks is presented. The Ant Colony Optimization (ACO) method is used to tune the parameters of PID controller subject to H-infinity constraints. The nonlinear dynamic model for multiple TCP flows control is developed based on fluid-flow theory. The designed controller provides good tracking performance in the presence of wide range of system parameter uncertainty. NS2 package is used to perform the nonlinear simulation of the system.

Keywords – Computer networks; Active Queue Management; Congestion control; Robust control; ACO; Uncertain systems.

[†] This paper has been presented in ECCCM-2 Conference and accredited for publication according to IJCCCE rules.

1. Introduction

With increase in dependence on networks like Internet, there is an increase in contention for the network's resources. This contention has affected the performance of networks. While any network performs reasonably well under light load, problems surface when they are used extensively. The most notable and common problem that networks are faced with is loss of data. While loss of data in a network can occur due to a variety of reasons, congestion in the network is the most common reason. Loosely speaking, congestion refers to the loss of network performance when a network is heavily loaded. This loss of performance can be data loss, large delays in data transmission, which is often unacceptable. Due to this, controlling and avoiding congestion is a critical problem in network management and design [1, 2]. Active Queue Management (AQM), as one class of packet dropping/marketing mechanism in the router queue, has been proposed to support the end-to-end congestion control in the Internet. The network congestion phenomenon is induced when the amount of data injected in the network is larger than the amount of the data which is delivered to destinations. The approach which represents the beginning of network congestion control is a so called end-to-end approach, where the responsive data sources reduce their transmission rate when they infer congestion occurrences from packet losses. This is, for instance, the approach adopted by the TCP protocol [3 - 5].

In this paper, two types of robust controllers are proposed in order to design robust AQM algorithms, these controllers are conventional H_∞ and robust ACOPID controllers. These controllers are designed

to achieve the system objectives in the presence of system uncertainties and/or disturbance.

2. TCP/AQM Model and Topology

A dynamic model of TCP behavior has been developed using fluid-flow and stochastic differential equation analysis. Ignoring the TCP timeout mechanism, a model is developed as in [6, 7]. The AQM nonlinear characteristics can be linearized about the operating point (nominal point) to yield the following linearized model and can be expressed as:

$$P(s) = \frac{W(s)}{p(s)} = \frac{\frac{C^2}{2N}}{(s + \frac{2N}{R_0^2 C})(s + \frac{1}{R_0})} \times \exp(-sR_0) \quad (1)$$

Where,

R_0 = round trip time;

C = link capacity (packets/sec.);

N = load factor (Number of TCP session);

The nominal plant is selected with the network parameter: $N=60$ TCP sources, $R_0=0.253$ sec. and $C=15$ Mbps (3750 packet/sec) as shown in Fig.1. The resulting transfer function of the nominal plant is:

$$P_0(s) = \frac{1.1725 \times 10^5}{s^2 + 4.452s + 1.976} \times \exp(-0.253s) \quad (2)$$

The values of the uncertain system parameters and their variation range are given in Table 1.

Table 1: The nominal system model parameters and their range

Uncertain parameter	$R_0(s)$	N	C(Mbps)
Minimum value	0.1	50	5
Nominal value	0.253	60	15
Maximum value	0.5	300	15

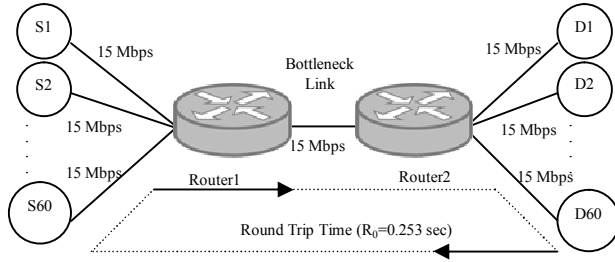


Figure 1 The conducted Network topology.

3. Ant Colony Optimization (ACO) method

ACO is a paradigm for designing meta-heuristic algorithms for combinatorial optimization problems. The essential trait of ACO algorithms is the combination of a priori information about the structure of a promising solution with information about the structure of previously obtained good solutions. ACO is founded on the foraging behavior of ants and their indirect communication based on pheromones, and has been applied to several combinatorial problems such as job scheduling and routing optimization in data. ACO's are especially suited for finding solutions to different optimization problems. A colony of artificial ants cooperates to find good solutions, which are an emergent property of the ant's cooperative interaction. Based on their similarities with ant colonies in nature, ant algorithms are adaptive and robust and can be applied to different versions of the same problem as well as to different optimization problems. If necessary in order to solve a particular optimization problem; artificial ants have been enriched with some additional capabilities not present in real ants. An ant searches collectively for a good solution to a given optimization problem. Each individual ant can find a solution or at least part of a solution to the optimization problem on its own but only when many ants work together they can find the optimal solution

[8 -10]. ACO is depending upon the pheromone matrix $\tau = \{\tau_{ij}\}$ for the construction of good solutions. The initial values of the pheromone are larger than zero

$$\text{Set } \tau_{ij} = \tau_0 \forall (i, j), \text{ where } \tau_0 > 0 \quad (3)$$

The probability $P_{ij}^A(t)$ of choosing a node j at node i is defined in the equation (4). At each generation of the algorithm, the ant constructs a complete solution using this equation, starting at source node.

$$P_{ij}^A(t) = \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}{\sum_{i,j \in T^A} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta}; i, j \in T^A; \quad (4)$$

where η_{ij} = Cost function. α and β are the constants that determine the relative influence of the pheromone values and the heuristic values on the decision of the ant respectively.

T^A = the path effectuated by the ant A at a given time.

The quantity of pheromone $\Delta\tau_{ij}$ on each path may be defined as

$$\tau_{ij}^A = \begin{cases} L^{min} & \text{if } i, j \in T^A \\ L^A & \\ 0 & \text{else} \end{cases} \quad (5)$$

where

L^A : The value of the objective function found by the ant A.

$$\tau_{ij} = \rho\tau_{ij}(t-1) + \sum_{A=1}^{NA} \Delta\tau_{ij}^A(t) \quad (6)$$

L^{min} : The best solution carried out by the set of the ants until the current iteration.

The pheromone evaporation is a way to avoid unlimited increase of pheromone trails.

Also, it allows the forgetfulness of the bad choices:

where

$\Delta\tau_{ij}^A$: The quantity of pheromone on each path.

NA : Number of ants.

ρ : The evaporation rate $0 < \rho \leq 1$

4. Robust AQM Design

In this paper, a robust controller (AQM) to control the congestion in computer network has been designed. The design of this controller is based on H_∞ technique which is a powerful technique to design a robust controller. In H_∞ control the performance to be optimized is defined in terms of the sensitivity function $S(s)$ and complementary sensitivity function $T(s)$. The unstructured multiplicative output description of the uncertainty is used; the resulting uncertainty model for the system parameters changes is [11, 12]:

$$W_m(s) = \frac{1.565 S^4 + 6.944 S^3 + 3.121 S^2 + 0.00285 S + 2.95 \times 10^{-6}}{S^4 + 6.314 S^3 + 3.633 S^2 + 0.00914 S + 4.394 \times 10^{-6}} \quad (7)$$

On the other hand, the problem is to regulate the output of the nominal plant to follow some given reference signal and to reject the disturbance by designing a controller $K(s)$. To improve the performance of the plant the following structure of the performance weighting function W_p is suggested [11]:

$$W_p(s) = \frac{(s/\sqrt{M_s} + \omega_b)^2}{(s + \omega_b \sqrt{e_{ss}})^2} \quad (8)$$

In this work the PID controller is conducted to design a robust AQM based

on ACO algorithm and H_∞ constraints. The PID controller is:

$$K(s) = k_p + \frac{k_i}{s} + k_d s \quad (9)$$

Ant Colony Optimization (ACO) method is used to tune the PID controller parameters subject to H_∞ constraints to achieve the required robustness of the network. The optimal parameters have been obtained using ACO method by minimizing the following cost function:

$$J_{min} = \|W_p S + W_m T\|_\infty + \frac{1}{M} \sum_{i=1}^M (e_i - \mu)^2 \quad (10)$$

Where

J : the cost function.

M : the length of error vector.

μ : the mean of the error

e : the error

The overall block diagram of the system with ACO tuning algorithm is shown in Fig. 2.

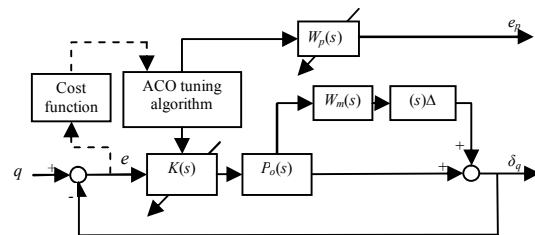


Figure 2: The overall block diagram of the controlled system.

The obtained PID controller and performance weighting function using ACO that covers the ranges of parameters change in Table 1 are:

$$K(s) = \frac{1.961 \times 10^{-5} S^2 + 7.529 \times 10^{-5} S + 3.33 \times 10^{-5}}{s} \quad (11)$$

$$W_p(s) = \frac{4.12 \times S^2 + 6.34 \times 10^5 S + 2800}{1.98 \times 10^3 S^2 + 1.92 \times 10^6 S + 16.9} \quad (12)$$

5. Results and Discussion

In this section, the performance and robustness of the proposed controller are evaluated by a number of simulations performed in MATLAB and NS2.

The weighting function parameters have been selected to obtain a minimum value of the closed loop. This can be checked by comparing the sensitivity function of the closed loop system with the inverse of the performance weighting function as shown in Fig. 3. From this figure, it is clear that the sensitivity function lies below the inverse of performance weighting function (W_p), which means that the performance criterion in Eq. (10) has been satisfied.

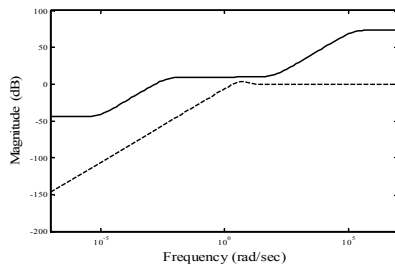


Figure 3 The resulting sensitivity function (*dotted*) compared with the inverse of the performance weighting function (*solid*).

The resulting complementary sensitivity function for the proposed controller has magnitudes less than the magnitudes of the inverse of the multiplicative uncertainty functions (W_m) as shown in Fig. 4. This means that the performance criterion in equation (10) has been satisfied for the proposed controller and the system is stable with all possible variations in system parameters. Furthermore, this means that all possible

variations in the network parameters are well covered by the determined uncertainty model.

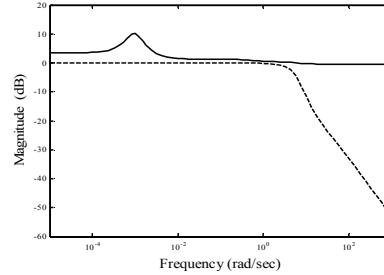


Figure 4 The resulting complementary sensitivity function (*dotted*) compared with the inverse of the uncertainty function (*solid*).

The time response of the system with the proposed controller is shown in Fig.5. From this figure, it can be seen that the controller can effectively compensate the system. The time response specifications obtained are: rise time is 0.4807sec., settling time is 0.9220sec. and with no overshoot.

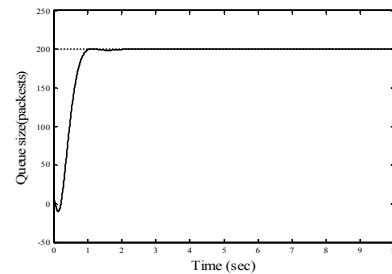


Figure 5 Step response characteristics of the uncertain system.

On the other hand, the nonlinear simulation is performed using NS2 package to verify the tracking performance of the proposed robust controller [13]. To evaluate the performance and robustness of the proposed controllers, there are many experiments are performed in this section by applying different values of the network parameters, different queue sizes and disturbances.

The system's stability and response when using the robust controller has been investigated under long-lived FTP flows. Each sender–receiver pair has TCP connections as cross traffic. The bottleneck link capacity and propagation time are fixed to 15 Mbps and 5msec. respectively. The numbers of TCP connections are selected to be changed as 100 and 300 TCP sources. Fig. 6 shows the performance of the network using the proposed controller with different values of TCP connections. From these figure, it is shown that the proposed robust controller can effectively stabilizing the network around the target in spite of the wide range of change in TCP connections. Furthermore, it is noted that the maximum peak overshoot is increased when the number of TCP sources is increased.

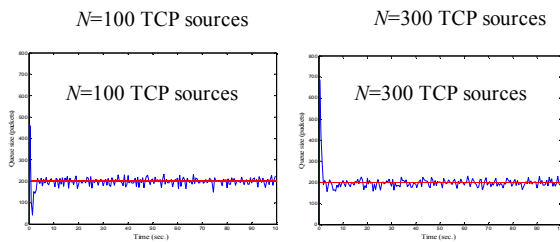


Figure 6 Network performance with different number of TCP sources, $C=15$ Mbps and $T_p=5$ msec.

The values of the mean and standard deviation of the queue length with the different number of TCP sources for the proposed robust controller in case $N=100$ and $N=300$ respectively are: mean =197.1, 198.9 and standard deviation = 34.9, 43.9. From these data, it can be concluded that the proposed robust controller can achieve desirable response.

The robustness of the proposed robust controller is investigated in a more dynamic situation by applying a sudden change in the number of the long-lived TCP flows. The performance of the network when the applied numbers of

TCP flows like as 60, 300 and 80 at time periods from 0-30, 30-60 and 60-100 sec. respectively are shown in Fig. 7. From this figure, it was shown that the proposed controller can robustly stabilize the system around the desired queue length (200 packets) with no losing in packets when a dynamic change of TCP connections is applied. The values of mean and standard deviation of the queue size using the proposed controller are: mean=197.6 and standard deviation=53.36.

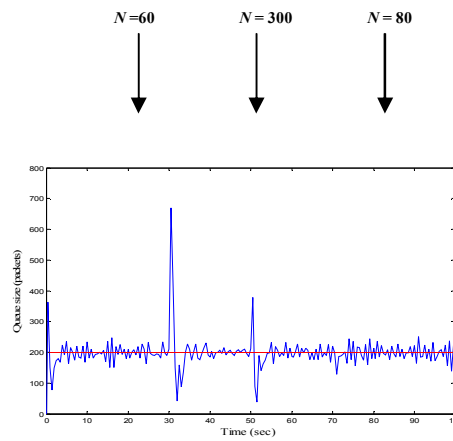


Figure 7 Network performance with different C, $N=100$ TCP sources and $T_p=5$ msec.

On the other hand, some of experiments have been done to investigate the ability of the proposed robust controller by varying the bottleneck link capacity. The performance of the proposed robust controller with different values of link capacity ($C = (5$ and $20)$ Mbps) is shown in Figure 8. The other parameters of the network are fixed to $N=100$ and $T_p = 5$ msec.. From this figure, it is clear that the proposed controller can robustly stabilizes the system around the target value (200 packets) in spite of the wide range of bottleneck link capacity change. The

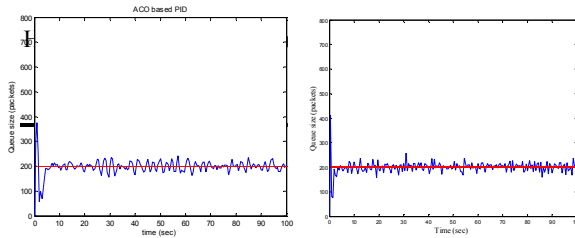


Figure 8: Network performance with different C , $N=100$ TCP sources and $T_p=5$ msec.

values of mean and standard deviation of the queue size using the proposed controller in case of $C=5$ Mbps and $C=20$ Mbps respectively are: mean=197.3, 199.7 and standard deviation=31.9 and 29.12.

File Transfer Protocol (FTP) powers one of the most fundamental Internet functions for the transferring the files between computers. Web developers use FTP protocols to upload/update their web sites and download other information. The ability of AQM with applying a disturbance represented by additional FTP connections has been investigated. Additional FTP flows with (100 and 500) nodes were added during the course of simulation. For the number of TCP connections is fixed to 100 and the additional FTP is added in the time interval between 20sec. and 40sec.. The network performance is shown in Fig.9. This figure shows that the proposed controller can keep the queue close to the desired queue length (200 packets) in the presence of the additional FTP connections and the system is robustly stable. The resulting mean and standard deviation data of the queue length using the proposed controller in case of additional 100 FTP connections and 500 FTP connections respectively are mean =198.7, 198.1 and standard deviation =33.85, 52.48.

Another type of external perturbation (disturbance) can be applied by adding additional unresponsive User Datagram Protocol (UDP) flows (25 and 60) nodes. Figure 10 shows the network performance with different UDP flows using the

proposed robust controller, it can be noted that the effect of a small number of unresponsive UDP flows for example (25 nodes) is equivalent to the effect of (300 nodes) FTP flows. The UDP flows start at 20sec. and stop at 40sec. and through this period the queue length with the robust controller is fluctuated within acceptable magnitude.

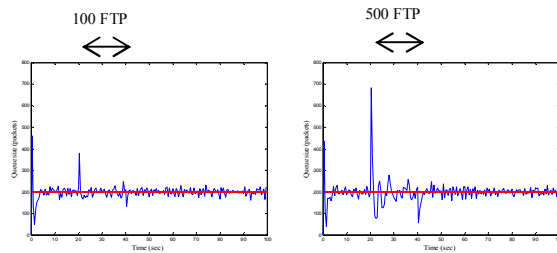


Figure 9: Network performance under the disturbance with the different additional FTP, $N=100$ TCP sources, $C=15$ Mbps and $T_p=5$ msec.

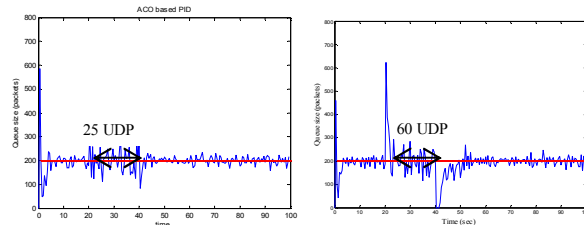


Figure 10: Network under the disturbance with the different additional UDP flows, $N=100$ TCP sources, $C=15$ Mbps and $T_p=5$ msec.

The resulting mean and standard deviation data of the queue length using the proposed controller in case of additional 25 UDP nodes and 60 UDP nodes respectively are mean =197.3, 194.4 and standard deviation =54.06, 59.76.

To investigate the ability of the proposed robust AQM scheme under varying the propagation delay, the following experiments are presented. The

performance of the controller with different T_p : (20 and 140) msec. is shown in Figure 11. The other network parameters are fixed to $N=100$ and $C=15$ Mbps. From this figure it was shown that the proposed controller can robustly stabilize the system around the target value (200 packets) for all values of T_p . The values of mean and standard deviation of the queue size using the proposed controller in case of $T_p=20$ msec. and $T_p=140$ msec. respectively are: mean=198, 188.9 and standard deviation = 35.48 and 45.84.

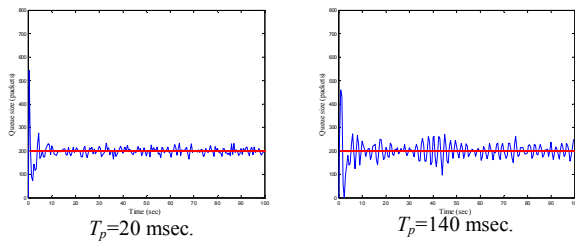


Figure 11: Network performance controller with different T_p , $N=100$ TCP sources and $C=15$ Mbps.

Table 2: Comparison between the performance of the proposed and the performance of the designed controllers in previous work.

Controller	Mean	Standard Deviation
PI controller [14]	220.68	45.59
Adaptive controller [15]	208.43	41.75
Proposed controller	195.6	33.09

Table 2 illustrates a comparison between the performance of the proposed robust controller and performance of the controllers in previous works in case of ($N=60$ TCP sources, $C=15$ Mbps, $T_p=5$ msec. and Queue size=200 packets).

6. Conclusions

In this paper a robust PID controller is proposed in order to design robust AQM algorithm. This controller is designed based on H_∞ technique to achieve the system objectives in the presence of system uncertainties and/or disturbance. The Ant Colony optimization method has been used to simplify the complexity of the design procedure of conventional H_∞ control and to achieve the same robustness with simpler structure and lower order controller. ACO algorithm has obtained the optimal parameters of the performance weighting function and PID controller that satisfies the robust stability and robust performance conditions (H_∞ constraints). This controller has achieved a time response performance better than that achieved by PI and adaptive controllers. The proposed controllers have been verified using MATLAB simulation and NS2 simulator. The superiority of robust ACOPID controller to the PI and adaptive controller was clear in terms of rise time = 0.408 sec, and settling time= 0.77 sec.

REFERENCES

- [1] D. E. Mcdysan and D. Paw, "ATM & MPLS Theory & Application: Foundations of Multi-Service Networking", McGraw-Hill, 2002.
- [2] A. S. Tanenbaum, "Computer Networks", 3th, Prentice hall of India Private Limited, 2002.
- [3] K. W. Ross, "Computer Networking a Top-Down Approach", sixth edition, ISBN-13: 978-0-13-285620-1, Polytechnic Institute of NYU, 2012.
- [4] V. Misra, W. B. Gong and D. Towsley, "Fluid-Based Analysis of a Network of AQM Routers Supporting TCP Flows with an Application to RED", in Proc. ACM Sigcomm, pp. 151-160, 2000.

-
- [5] S. Floyd and V. Jacobson, "Random Early Detection Gateway for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, pp. 397-413, August, 1993.
- [6] S. Floyd and V. Jacobson, "Random Early Detection Gateway for Congestion Avoidance", *IEEE/ACM Transactions on Networking*, Vol. 1, No. 4, pp. 397-413, August, 1993.
- [7] C. V. Hollot, V. Misra, D. Towsley and Wei-Bo Gong, "A Control Theoretic Analysis of RED", *Proceedings of IEEE, Infocom*, Vol. 3, pp. 1510-1519, 2001.
- [8] I. Chiha, N. Liouane and P. Borne, "Tuning PID Controller Using Multiobjective Ant Colony Optimization", *Hindawi Publishing Corporation, applied Computational Intelligence and Soft Computing*, Vol. 2, 2012.
- [9] B. Nagaraj and P. Vijayakumar, "A Comparative Study of PID Controller Tuning using GA, EP, PSO and ACO", *Journal of Automation, Mobile Robotics & Intelligent Systems Articles*, Vol. 5, No. 2, 2011.
- [10] I. Chiha and P. Borne, "Multi-Objective Ant Colony Optimization to Tuning PID Controller", *Applied Computational Intelligence and Soft Computing*, 2012.
- [11] K. Zhou and J. C. Doyle, "Essentials of Robust Control", *Prentice Hall*, New Jersey, 1998.
- [12] F. Lin, "Robust Control Design: An Optimal Control Approach", *John Wiley & Sons Ltd*, England, 2007.
- [13] The NS Manual, "Formerly NS Notes and Documentation", A Collaboration between Researchers at UC Berkeley, LBL, USC/ISI, and Xerox Parc, January, 2009.
- [14] C. V. Hollot, V. Misra, D. Towsley and W. B. Gong, "On Designing Improved Controllers for AQM Routers Supporting TCP Flows", *In Proceedings of IEEE In Focom, USA*, pp. 1726-1734, 2001.
- [15] H. J. Hassan, "Development of Adaptive Wavelet Network PID (AWNPID) controller for improving AQM in the TCP network", *Ph.D. thesis, University of Technology, Iraq*, 2013.